

1   **What Is Claimed:**

2   1.     A system for reducing latency of computer operations, comprising:

3         a first processing pipeline comprising a prevalidated cache translation lookaside  
4         buffer (TLB), the prevalidated cache TLB comprising a virtual address (VA) content  
5         addressable memory (CAM), wherein the VA CAM receives virtual address information  
6         for integer load operations; and

7         a second processing pipeline, independent of the first processing pipeline, the  
8         second processing pipeline comprising:

9             a cache tag array that holds physical addresses of cache lines,  
10            a master TLB that receives virtual address information for store operations  
11            and generates a physical address,  
12            a bypass around the master TLB, wherein if the store address is a physical  
13            address, the physical address bypasses the master TLB, and  
14            a comparator that compares a physical address from one of the bypass and  
15            the master TLB to a physical address from the cache tag array, wherein if the physical  
16            addresses match, a store/invalidate cache way hit is generated.

17   2.     The system of claim 1, wherein the first processing pipeline further comprises:  
18         a data cache;

19         prevalidated cache tag array; and

20         a logic circuit at the output of the VA CAM and the prevalidated cache tag array  
21         to produce a cache way hit.

22   3.     The system of claim 2, wherein the first processing pipeline further comprises  
23         a multiplexor that uses the cache way hit to select a data cache output.

24   4.     The system of claim 1, wherein the first processing pipeline further comprises  
25         a column clear module that clears TLB hit bits of the prevalidated cache tag array for one  
26         or more TLB slot entry positions.

27   5.     The system of claim 1, wherein the prevalidated cache TLB sets TLB hit bits for  
28         all TLB slots to be invalidated.

29   6.     The system of claim 1, further comprising a hardware cleansing function that  
30         removes stale cache lines after each prevalidated cache TLB insert or purge, wherein the  
31         function:

32             compares each cache tag index that has an invalid entry in the prevalidated  
33             cache tags with a valid entry in the physical address cache tags; and

1                   clears each physical address cache tag when the comparison results in a  
2 match.

3       7.       The system of claim 1, further comprising:

4           an apparatus to invalidate store valid bits whenever a TLB bit in the prevalidated  
5 cache tag array is cleared due to a column clear operation, wherein the apparatus:

6                   detects a column clear function;

7                   ORs the multiple bits; and

8                   compares each bit invalidated in the prevalidated cache tag array to each  
9 index row of the store valid bits in both the prevalidated cache tag array and the physical  
10 address cache tag.

11       8.       The system of claim 1, further comprising:

12           a stale cache line control to invalidate store valid bits on one index location when  
13 an index is being loaded with a new cache line, wherein the stale cache line control:

14                   receives cache way hit information and, optionally, column clear  
15 information;

16                   provides signals to a store valid bit module located in both the first  
17 processing pipeline and the second processing pipeline;

18                   sends the address of a cache line that is being processed in a cache line fill  
19 through the physical address cache tag;

20                   determines if the cache line is stale in any of the cache ways that are not  
21 being loaded with the new line whenever there is a hit on any of the other cache ways;

22                   invalidates the stale cache line; and

23                   clears the store valid bits in both store valid bit modules.

24       9.       A method for reducing latency of computer operations, comprising:

25           running a first processing pipeline wherein the first processing pipeline receives  
26 virtual address information for integer load operations from a pre-validated cache TLB  
27 with a VA CAM; and

28           running a second processing pipeline, independent of the first processing pipeline,  
29 wherein the second processing pipeline:

30                   holds physical addresses of cache lines in a physical address cache tag  
31 array;

32                   receives address information for store operations in a master TLB, wherein  
33 the master TLB generates a physical address;

34                   bypasses around the master TLB if the store address is a physical address;

1 compares the physical address from one bypass and the master TLB to a  
2 physical address from the cache array; and  
3 generates a cache way hit if the physical addresses match.

4 10. The method of claim 9, further comprising:  
5 clearing entries from the prevalidated cache tag array for one or more prevalidated  
6 cache TLB slot entry positions.

7 11. The method of claim 9, further comprising:  
8 invalidating prevalidated cache tags by:  
9 setting prevalidated cache TLB hit bits for all prevalidated cache TLB  
10 slots that are to be invalidated;  
11 receiving a column clear signal by the prevalidated cache tag array; and  
12 clearing all prevalidated cache TLB hit bits in the prevalidated cache tag  
13 array for all TLB slots indicated by the TLB hit signals.

14 12. The method of claim 9, further comprising:  
15 removing cache lines that are invalidated by a column clear but remain valid for a  
16 stale cache line by:  
17 comparing each cache tag index that has an invalid entry in the  
18 prevalidated cache tag array with a valid entry in the physical address cache tag array;  
19 and  
20 clearing each physical address cache tag when the comparison results in a  
21 match.

22 13. The method of claim 9, further comprising:  
23 detecting a column clear function;  
24 connecting the column clear function information to each index row of the store  
25 valid bits in both the prevalidated cache tag array and the physical address cache tag  
26 array.

27 14. The method of claim 13, further comprising OR-ing multiple store valid bits.

28 15. The method of claim 9, further comprising:  
29 removing stale cache lines by:  
30 receiving cache way hit information and optionally column clear  
31 information;  
32 providing signals to a store valid bit module located within the first  
33 processing pipeline and the second processing pipeline;

1                    sending the address of a cache line that is being processed in the cache  
2    through the physical address cache tag array;  
3                    determining if the cache line is stale in any of the cache ways that are not  
4    being loaded with the new line whenever there is a hit on any of the other cache ways;  
5    and  
6                    invalidating the stale cache line and clearing the store valid bits in the store  
7    valid bit module located in the first processing pipeline and the second processing  
8    pipeline.